

Reduced Basis Method for Finite Volume Approximations of Parametrized Evolution Equations

Bernard Haasdonk and Mario Ohlberger

November 15, 2006

Abstract

The model order reduction methodology of *reduced basis (RB)* techniques offers efficient treatment of parametrized partial differential equations (P²DEs) by providing both approximate solution procedures and efficient error estimates. RB-methods have so far mainly been applied to finite element schemes for elliptic and parabolic problems. In the current study we extend the methodology to general evolution schemes such as finite volume schemes for parabolic and hyperbolic evolution equations. The new theoretic contributions are the formulation of a reduced basis approximation scheme for general evolution problems and the derivation of rigorous a-posteriori error estimates in various norms. Algorithmically, an offline/online decomposition of the scheme and the error estimators is realized. This is the basis for a rapid online computation in case of multiple-simulation requests. We introduce a new offline basis-generation algorithm based on our a posteriori error estimator which combines ideas from existing approaches. Numerical experiments for an instationary convection-diffusion problem demonstrate the efficient applicability of the approach.

1 Introduction

Many real world problems can be modelled by parametrized partial differential equations (P²DEs), where a parameter vector $\boldsymbol{\mu}$ characterizes the system in terms of material-, geometry- or control-parameters. In simulation of general P²DEs, repeated computation runs for varying parameters $\boldsymbol{\mu}$ and fast simulation response can be required. Such scenarios occur in design optimization, optimal control with PDE-constraints, online-simulation, parameter identification or state estimation. Model order reduction techniques must be applied to satisfy these time demands. The methodology of *reduced basis (RB)* techniques aims at efficient treatment in such cases by providing both an approximate solution procedure and efficient error estimates.

In the current study we extend the methodology to general parametrized evolution equations of the form

$$\partial_t u(t; \boldsymbol{\mu}) + L[u(t; \boldsymbol{\mu}); \boldsymbol{\mu}] = 0$$

with suitable initial data $u(0; \boldsymbol{\mu}) = u_0(\boldsymbol{\mu})$ and boundary conditions. Here L denotes an operator which comprises the spatial derivatives, $u(\boldsymbol{\mu}) : [0, T_{\max}] \rightarrow \mathcal{W}$ is the desired parameter-dependent solution and \mathcal{W} is a subset of $L^2(\Omega)$ or some other suitable Hilbert-space of functions on a domain Ω . Such evolution equations are frequently solved by numerical evolution schemes. These do not only comprise finite element methods (FEM), which represent the main focus in the RB-literature. In particular, our framework covers finite volume (FV) schemes [4, 8, 12, 13] for parabolic and hyperbolic transport equations. Similarly, local discontinuous Galerkin (LDG) methods [2, 6, 5] can be cast in this framework. Discrete evolution schemes are based on an approximating high-dimensional discrete space $\mathcal{W}_H \subset L^2(\Omega)$ (or subset of some other Hilbert space) and by approximating the exact solution at discrete times-instants t_k , i.e. providing a sequence of functions $u_H^k(\boldsymbol{\mu}) \in \mathcal{W}_H$ for $k = 0, \dots, K$ such that $u_H^k(\boldsymbol{\mu}) \approx u(t_k; \boldsymbol{\mu})$. The discrete solution $u_H(\boldsymbol{\mu})$ then is defined to

be piecewise constant in time with $u_H(t_k; \boldsymbol{\mu}) = u_H^k(\boldsymbol{\mu})$. The evolution scheme itself proceeds by setting $u_H^0(\boldsymbol{\mu}) := P[u_0(\boldsymbol{\mu})]$, where P is a suitable projection operator, and iteratively solving $L_I^k[u_H^{k+1}(\boldsymbol{\mu})] = L_E^k[u_H^k(\boldsymbol{\mu})] + b^k(\boldsymbol{\mu})$ for $u_H^{k+1}(\boldsymbol{\mu})$. Here L_I^k and L_E^k denote ($\boldsymbol{\mu}$ -dependent) implicit and explicit contributions gathering the discretization of the time derivative and spatial differential operator L at time t_k .

The aim of RB-methods is a fast online-phase, in which for a newly given parameter vector $\boldsymbol{\mu}$ a function $u_N(\boldsymbol{\mu})$ is determined by a Galerkin projection such that it approximates the unknown $u_H(\boldsymbol{\mu})$, and the error can be quantified or bounded. These computations ideally have a complexity independent of the dimensionality H of the original non-reduced model. Instead, the reduced model is based on a very low dimensional approximate space $\mathcal{W}_N \subset \mathcal{W}_H$, the reduced basis space with $\dim \mathcal{W}_N = N$. This approximating subspace is determined in an extensive offline-phase, such that it finally best captures the solution under all possible parameter variations.

Similar to related studies, the RB-approach will be developed by requiring affine parameter dependence in the data functions and linearity in the evolution equation and evolution scheme. This enables a complete offline/online decomposition which is the basis for the efficient reduced approximation scheme and rigorous a-posteriori error control. Although we restrict to the linear setting in the current presentation, this is an important step for the general nonlinear case. The treatment of nonlinearities, which can be caused by nonlinear data functions or nonlinear evolution operators, will similarly rely on linearizations and approximate affine decompositions.

RB-methods have been applied widely in the last decades ranging back to early works with applications in structural statics, such as [1] where *global shape functions* are used for spanning the approximating space, or [18], where the notion *reduced basis technique* can already be found. Application of reduced basis methods on time-dependent ordinary differential equations and corresponding a-priori error analysis can be found in [21]. These methods have mainly been adopted to finite element schemes for elliptic and parabolic problems. In recent years the RB-methods have gained popularity and developed to a rather extensive methodology. Studies on eigenvalue problems are treated with RB-methods in [14]. The general methodology for linear elliptic equations is the subject of [23], details on algorithmic aspects can be found in [22]. The stationary advection-diffusion problem has been treated in [24]. The methodology has been successively extended in various directions. Time dependency of problems is addressed for linear parabolic equations in [10]. The treatment of polynomial nonlinearities can be realized by interpreting the nonlinearity as a multilinear form, e.g. in case of the stationary viscous Burgers equation [29]. An overview of the general methodology for both linear and multilinear cases can be found in [17]. Non-affine parameter dependencies can be handled by approximation methods such as the *empirical interpolation* scheme [3], which also is the basis for more general nonlinearities [9]. The RB-methodology is also applicable to systems of equations such as the incompressible Navier-Stokes equations [28]. In combination with geometry optimization the framework is developed in [26]. More complex systems of algebro-differential equations can be treated by Galerkin projection approaches. For instance model reduction of a fuel-cell is successfully performed in [15].

The structure of the present study is as follows: In the next sections we introduce the notation and terminology for finite volume schemes. An abstraction into a general implicit/explicit operator formulation is performed. In Sec. 3 the reduced basis technique is transferred to a class of general time evolution schemes. Different a-posteriori error estimates are presented in the subsequent section. Under certain assumptions a complete independence of the dimensionality of the discrete space can be obtained during the reduced basis simulation. This is realized by the offline/online decomposition presented in Sec. 5. The subsequent section presents a new algorithm for reduced basis construction. Numerical experiments in Sec. 7 demonstrates the applicability of the approach on a transport problem which (largely simplified) models a fuel cell component. We close with concluding remarks in the last section.

2 Finite Volume Approximation

The general model of the P²DE under consideration is the following parametrized evolution equation with general initial data and boundary conditions. Let $\Omega \subset \mathbb{R}^d$ denote some bounded polygonal space domain with boundary $\partial\Omega = \Gamma_{\text{dir}} \cup \Gamma_{\text{neu}}$ decomposed into Dirichlet and Neuman components and $[0, T_{\text{max}}]$ be a time interval. For any parameter vector $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^p$ the function $u(\mathbf{x}, t; \boldsymbol{\mu})$ denotes the solution of the parametrized advection-diffusion equation:

$$\partial_t u(\boldsymbol{\mu}) + \nabla \cdot (\mathbf{v}(\boldsymbol{\mu})u(\boldsymbol{\mu}) - d(\boldsymbol{\mu})\nabla u(\boldsymbol{\mu})) = 0 \quad \text{in } \Omega \times [0, T_{\text{max}}], \quad (1)$$

$$u(\cdot, 0; \boldsymbol{\mu}) = u_0(\boldsymbol{\mu}) \quad \text{in } \Omega, \quad (2)$$

$$u(\boldsymbol{\mu}) = b_{\text{dir}}(\boldsymbol{\mu}) \quad \text{in } \Gamma_{\text{dir}} \times [0, T_{\text{max}}], \quad (3)$$

$$(\mathbf{v}(\boldsymbol{\mu})u(\boldsymbol{\mu}) - d(\boldsymbol{\mu})\nabla u(\boldsymbol{\mu})) \cdot \mathbf{n} = b_{\text{neu}}(u; \boldsymbol{\mu}) \quad \text{in } \Gamma_{\text{neu}} \times [0, T_{\text{max}}]. \quad (4)$$

In particular, the initial data u_0 is space-dependent, the velocity field \mathbf{v} , the diffusion coefficient d , the boundary value function b_{dir} and b_{neu} may be space and time dependent. The Neuman-boundary conditions are assumed to be affine in u by $b_{\text{neu}}(u, \mathbf{x}, t; \boldsymbol{\mu}) = b_{\text{neu},1}(\mathbf{x}, t; \boldsymbol{\mu})u(\mathbf{x}, t; \boldsymbol{\mu}) + b_{\text{neu},0}(\mathbf{x}, t; \boldsymbol{\mu})$. This covers usual flow-conditions such as noflow or outflow. The geometry and data is assumed to be sufficiently regular, such that the solution is well-defined.

Finite volume schemes are frequently used instead of finite elements for such partial differential equations. Firstly, they enable to model discontinuous solutions, which can occur in the hyperbolic case and secondly they are motivated by and reflect the conservativity of the problem. Therefore, we focus on the following discrete function space for such equations.

Definition 2.1 (Finite Volume Space).

- Let $0 = t_0 < t_1 < \dots < t_K = T_{\text{max}}$ be a sequence of time instants with corresponding time-steps $\Delta t_k := t_{k+1} - t_k$ for $k = 1, \dots, K - 1$.
- $\mathcal{T} = \{T_i\}_{i=1}^H$ denotes a convex polygonal tessellation of Ω , i.e. a collection of convex polygons with $\bigcup T_i = \bar{\Omega}$, $T_i \cap T_j = \emptyset$ for $i \neq j$.
- The faces of an element T_i are denoted as e_{ij} with j stemming from a corresponding index set $j \in \mathcal{N}(i) := \mathcal{N}_{\text{in}}(i) \cup \mathcal{N}_{\text{dir}}(i) \cup \mathcal{N}_{\text{neu}}(i)$ consisting of the disjoint index sets of inner, Dirichlet and Neuman boundary segments.
- The center of gravity of T_i is denoted with \mathbf{c}_i and \mathbf{c}_{ij} denotes the center of gravity of the face e_{ij} . The volume of such entities is denoted with $|T_i|$, $|e_{ij}|$.
- Points $\mathbf{s}_i \in T_i$ are assumed to exist, such that $\mathbf{s}_i \mathbf{s}_j$ is perpendicular to e_{ij} , the intersection point is denoted as \mathbf{s}_{ij} .
- $\mathcal{W}_H := \text{span}(\chi_{T_i}) \subset L^\infty(\Omega) \subset L^2(\Omega)$ is the discrete space of cell-wise constant functions, where χ_{T_i} are the characteristic functions on the elements T_i . The space is equipped with the scalar product $\langle \cdot, \cdot \rangle$ and norm $\|\cdot\|$ inherited from $L^2(\Omega)$.

In the space \mathcal{W}_H , which is usually of high dimension $H = \dim(\mathcal{W}_H)$, the corresponding finite volume schemes can be formulated. We allow general inhomogeneous boundary conditions and mixed implicit/explicit contributions.

Definition 2.2 (Explicit/Implicit Form of FV-Approximation).

- $u_H(\mathbf{x}, t; \boldsymbol{\mu})$ denotes the overall finite volume (FV) approximation, which is defined to coincide with $u_H^k(\mathbf{x}; \boldsymbol{\mu})$ in the time-slab $[t_k, t_{k+1})$.
- $u_H^0(\boldsymbol{\mu}) := P[u_0(\boldsymbol{\mu})] \in \mathcal{W}_H$ is the L^2 -projection of the initial data, i.e.

$$u_H^0(\mathbf{x}; \boldsymbol{\mu}) := u_i^0 := \frac{1}{|T_i|} \int_{T_i} u_0(\cdot; \boldsymbol{\mu}) \quad \text{for } \mathbf{x} \in T_i. \quad (5)$$

- $u_H^{k+1}(\boldsymbol{\mu}) \in \mathcal{W}_H$ is the function obtained by a time evolution step defined by

$$u_H^{k+1}(\mathbf{x}; \boldsymbol{\mu}) := \sum_i \chi_{T_i} u_i^{k+1}, \quad (6)$$

where the $u_i^{k+1} \in \mathbb{R}$ satisfy

$$u_i^{k+1} = u_i^k - \frac{\Delta t_k}{|T_i|} \sum_{j \in \mathcal{N}(i)} h_{ij}^k(u_H^k, u_H^{k+1}; \boldsymbol{\mu}). \quad (7)$$

- $h_{ij}^k(u, v; \boldsymbol{\mu})$ denotes a numerical flux, discretizing the convective and diffusive flux from element i over the edge e_{ij} if $u, v \in \mathcal{W}_H$ are the functions at time t_k and t_{k+1} respectively.

Under specific non-degeneracy requirements on the tessellation, consistency requirements between numerical fluxes and analytical data coefficients and a stability bound on the time-step size, such schemes can be proven to converge to the unique entropy solution (see e.g. [7] and the references therein).

The above formulation covers a wide range of numerical fluxes discretizing the convective and diffusive part explicitly, implicitly, or in a mixed way, and includes higher order discretizations by using limiters [12]. For the convective part usual choices are monotone fluxes such as the Lax-Friedrichs, Enquist-Osher or the Godunov flux [12, 4]. For the diffusive flux simple difference approximations of the gradient can be performed [19, 12], occasionally involving harmonic weighting of cell-averaged diffusivities [11], or more general gradient reconstruction techniques [7].

For instance, in the experiments we use the explicit Lax-Friedrichs-flux for the advective contribution, an implicit discretization of the diffusivity and suitable quadrature rules:

$$h_{ij}^k(u_H^k, u_H^{k+1}; \boldsymbol{\mu}) = g_{ij}^k(u_H^k; \boldsymbol{\mu}) + d_{ij}^k(u_H^{k+1}; \boldsymbol{\mu}), \quad (8)$$

$$g_{ij}^k(u_H^k; \boldsymbol{\mu}) = \frac{1}{2}|e_{ij}| \left(\mathbf{v}(\mathbf{c}_{ij}) \cdot \mathbf{n}_{ij} (u_j^k + u_i^k) - \frac{1}{\lambda} (u_j^k - u_i^k) \right), \quad (9)$$

$$d_{ij}^k(u_H^{k+1}; \boldsymbol{\mu}) = -d(\mathbf{s}_{ij}) \frac{|e_{ij}|}{|\mathbf{s}_i - \mathbf{s}_j|} (u_j^{k+1} - u_i^{k+1}). \quad (10)$$

Here λ denotes a coefficient which introduces a stabilizing artificial viscosity. This choice of fluxes can be applied for all inner edges yielding h_{ij}^k for $j \in \mathcal{N}_{\text{in}}(i)$. The Neuman boundary contribution can be respected by a face-integral of the Neuman-boundary-values $h_{ij}^k := \int_{e_{ij}} b_{\text{neu}}$. The Dirichlet values at a face $e_{ij} \subset \Gamma_{\text{dir}}$ of a boundary element T_i can be incorporated via constructing a ghost cell T_j outside of Ω by reflecting T_i with respect to the boundary face e_{ij} , assigning the Dirichlet-values $u_j^k := |e_{ij}|^{-1} \int_{e_{ij}} b_{\text{dir}}$ to these elements and computing the numerical flux h_{ij}^k as defined for inner cells (cf. [20]).

Overall, the solution-dependent quantities can be collected into implicit and explicit operators $L_I^k(\boldsymbol{\mu})$ and $L_E^k(\boldsymbol{\mu})$ and solution-independent functions $b^k(\boldsymbol{\mu})$ generated by non-homogeneous boundary values. For this, we define the general class of evolution schemes, for which the RB-approximation and error-estimates will be derived.

Definition 2.3 (Parametrized Evolution Scheme). *We denote a numerical scheme, which produces $\boldsymbol{\mu}$ -dependent solutions $u_H^k(\boldsymbol{\mu}) \in \mathcal{W}_H$ for $k = 0, \dots, K$ a parametrized evolution scheme, if there exist continuous linear operators $P : L^2(\Omega) \rightarrow \mathcal{W}_H$, $L_I^k(\boldsymbol{\mu}) : \mathcal{W}_H \rightarrow \mathcal{W}_H$, $L_E^k(\boldsymbol{\mu}) : \mathcal{W}_H \rightarrow \mathcal{W}_H$, functions $b^k(\boldsymbol{\mu}) \in \mathcal{W}_H$ and a constant $C_E > 0$ where L_E^k satisfies $\|L_E^k[\varphi]\| \leq C_E \|\varphi\|$ for all $\varphi \in \mathcal{W}_H$ and L_I^k can be decomposed as $L_I^k = (Id + \Delta t_k \bar{L}_I^k)$ with positive semidefinite \bar{L}_I^k such that the scheme can be written as*

$$u_H^0 = P[u_0(\boldsymbol{\mu})], \quad (11)$$

$$L_I^k(\boldsymbol{\mu})[u_H^{k+1}] = L_E^k(\boldsymbol{\mu})[u_H^k] + b^k(\boldsymbol{\mu}). \quad (12)$$

As mentioned in the introduction, this definition captures a variety of discretizations, in particular the FV-discretization of the advection-diffusion equation. The assumptions on the operators can be realized for instance by the above given choice of fluxes and arbitrary quadratures for the boundary integrals. The proof of the following statement is omitted, as it can be obtained easily from the definitions.

Proposition 2.4 (FV-Approximation as Parametrized Evolution Scheme). *The finite volume discretization according to Def. 2.2 of the parametrized advection-diffusion equation with fluxes (8)-(10) can be written as a parametrized evolution scheme in the sense of Def. 2.3 by $P : \mathcal{W} \rightarrow \mathcal{W}_H$ being the orthogonal $L^2(\Omega)$ -projection, and defining the operators L_I^k, L_E^k by their corresponding matrices and the functions b^k by their coefficients (with respect to the basis $\{\chi_{T_i}\}$):*

$$\left(L_I^k\right)_{il} := \begin{cases} 1 + \frac{\Delta t_k}{|T_i|} \sum_{j \in \mathcal{N}_{\text{in}}(i) \cup \mathcal{N}_{\text{dir}}(i)} d(\mathbf{s}_{ij}) \frac{|e_{ij}|}{|\mathbf{s}_i - \mathbf{s}_j|} & \text{for } l = i, \\ -\frac{\Delta t_k}{|T_i|} d(\mathbf{s}_{il}) \frac{|e_{il}|}{|\mathbf{s}_i - \mathbf{s}_l|} & \text{for } l \in \mathcal{N}_{\text{in}}(i), \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

$$\left(L_E^k\right)_{il} := \begin{cases} 1 - \frac{\Delta t_k}{|T_i|} \left(\sum_{j \in \mathcal{N}_{\text{in}}(i) \cup \mathcal{N}_{\text{dir}}(i)} \frac{|e_{ij}|}{2} [\mathbf{v}(\mathbf{c}_{ij}) \cdot \mathbf{n}_{ij} + \lambda^{-1}] \right. \\ \quad \left. + \sum_{j \in \mathcal{N}_{\text{neu}}(i)} |e_{ij}| b_{\text{neu},1}(\mathbf{s}_{ij}) \right) & \text{for } l = i, \\ -\frac{\Delta t_k}{|T_i|} \frac{|e_{il}|}{2} [\mathbf{v}(\mathbf{c}_{il}) \cdot \mathbf{n}_{il} - \lambda^{-1}] & \text{for } l \in \mathcal{N}_{\text{in}}(i), \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

$$\left(b^k\right)_i := \begin{cases} -\frac{\Delta t_k}{|T_i|} \left(\sum_{j \in \mathcal{N}_{\text{dir}}(i)} \left(\frac{|e_{ij}|}{2} [\mathbf{v}(\mathbf{c}_{ij}) \cdot \mathbf{n}_{ij} - \lambda^{-1}] b_{\text{dir}}(\mathbf{c}_{ij}) \right. \right. \\ \quad \left. \left. - d(\mathbf{s}_{ij}) \frac{|e_{ij}|}{|\mathbf{s}_i - \mathbf{s}_j|} b_{\text{dir}}(\mathbf{s}_{ij}) \right) \right. \\ \quad \left. + \sum_{j \in \mathcal{N}_{\text{neu}}(i)} (|e_{ij}| b_{\text{neu},0}(\mathbf{c}_{ij})) \right). \end{cases} \quad (15)$$

Here, the data functions of the convective and Neuman-boundary contributions are evaluated at times t_k , the diffusive parts at time t_{k+1} .

3 Reduced Basis Approximation

In the following, the goal is to provide an efficient scheme for approximating the solutions $u_H^k \in \mathcal{W}_H$ from a parametrized evolution scheme with approximations $u_N^k \in \mathcal{W}_N$ in a lower dimensional space and provide a-posteriori error control. The FV-discretization will be used in the experiments as one example of such schemes, but the approach is more general. It is identically valid for, e.g. finite difference, finite element space discretization, staggered schemes, etc.

We denote the solution $u_H(\boldsymbol{\mu})$ from the evolution scheme as a *high-resolution* or *detailed* solution in contrast to the *reduced basis* solution $u_N(\boldsymbol{\mu})$ to be defined in this section.

We do not target at the error $u - u_H$ to the true analytical solution, so we assume, that the discretization is chosen adequately fine, i.e. H sufficiently high, such that the discretization error $u - u_H$ is negligible compared to the approximation error $u_N - u_H$.

The computational complexity of a single time-step in a parametrized evolution scheme is of order $\mathcal{O}(H^\gamma)$ for some $\gamma \geq 1$. In case of a full explicit discretization, i.e. $L_I^k = Id$, the exponent is typically $\gamma = 1$ for the single matrix multiplication. For discretizations with implicit contributions, we typically obtain $1 \leq \gamma \leq 3$ for the solution of the linear system. For a complete solution u_H these computational demands scale with the number of time-steps K , which in turn can be estimated in more detail depending on H and the dimensionality of the space \mathbb{R}^d . Irrespective of these details, the complexities are in any case highly dependent on H , so not suitable for an online simulation setting for very large values of H . This is the general motivation for reduced basis (RB) methods: The goal is an approximate solution strategy, which renders independent of the dimension H of

the high-dimensional space \mathcal{W}_H . Instead, a low dimensional, problem specific *reduced basis space* \mathcal{W}_N with dimension N is constructed based on particular precomputed high-resolution numerical solutions at certain times, so called snapshots. Based on \mathcal{W}_N the original numerical scheme is approximated.

Definition 3.1 (Reduced Basis Space).

- Let $\mathcal{S} := \{(\boldsymbol{\mu}_n, t_{k_n})\}_{n=1}^S$ be a sample of parameters from $\mathcal{P} \times \{t_k\}_{k=0}^K$.
- $\mathcal{W}_N \subset \text{span}\{u_H(\cdot, t_{k_n}; \boldsymbol{\mu}_n) | (\boldsymbol{\mu}_n, t_{k_n}) \in \mathcal{S}\}$ denotes an N -dimensional subspace of \mathcal{W}_H , the reduced basis space.
- $\Phi_N := \{\varphi_n\}_{n=1}^N$ is an orthonormalized basis of \mathcal{W}_N , a so called reduced basis.

Note, that we do not use snapshots directly as basis-functions but use orthonormal bases instead, as these are beneficial for the stability of the numerics [26]. Further, we do not take \mathcal{W}_N as a space spanned by snapshots, but as a subspace of such. By this choice, a compact basis-construction is possible. Details on the algorithms to construct \mathcal{W}_N are given in Sec. 6.

The next step in the RB-methodology is dimension reduction by a Galerkin-projection of the problem onto the reduced space. This means rewriting the parametrized evolution scheme in a weak form and restricting both the solution and the test-functions to the space \mathcal{W}_N . More precisely, functions $\{u_N^k(\boldsymbol{\mu})\}_{k=0}^K$ are wanted such that for all $\varphi \in \mathcal{W}_N$ and all $k = 0, \dots, K-1$ holds

$$\int_{\Omega} (u_N^0 - P[u_0(\boldsymbol{\mu})])\varphi = 0 \quad \text{and} \quad (16)$$

$$\int_{\Omega} (L_I^k(\boldsymbol{\mu})[u_N^{k+1}] - L_E^k(\boldsymbol{\mu})[u_N^k] - b^k(\boldsymbol{\mu}))\varphi = 0. \quad (17)$$

By expanding the RB-solution into the RB-basis and by using orthonormality of the basis $\{\varphi_n\}_{n=1}^N$ and linearity of the operators P , L_E^k and L_I^k , we obtain the following definition of the RB-approximation scheme for parametrized evolution equations.

Definition 3.2 (Explicit/Implicit Form of RB-Approximation). *We assume to have a parametrized evolution scheme as in Def. 2.3. Let $\{u_N^k(\boldsymbol{\mu})\}_{k=0}^K \subset \mathcal{W}_N$ be a sequence of approximate solutions such that $u_N^k(\mathbf{x}; \boldsymbol{\mu}) := \sum_{n=1}^N a_n^k(\boldsymbol{\mu})\varphi_n(\mathbf{x})$, where the coefficient vectors $\mathbf{a}^k := (a_n^k)_{n=1}^N$ satisfy*

$$\mathbf{a}^0 = \left(\int_{\Omega} P[u_0(\boldsymbol{\mu})]\varphi_1, \dots, \int_{\Omega} P[u_0(\boldsymbol{\mu})]\varphi_N \right)^T, \quad (18)$$

$$\mathbf{L}_I^k(\boldsymbol{\mu})\mathbf{a}^{k+1} = \mathbf{L}_E^k(\boldsymbol{\mu})\mathbf{a}^k + \mathbf{b}^k(\boldsymbol{\mu}), \quad (19)$$

and the required matrices and vectors are defined by the entries

$$(\mathbf{L}_I^k(\boldsymbol{\mu}))_{nm} := \int_{\Omega} \varphi_n L_I^k(\boldsymbol{\mu})[\varphi_m], \quad (\mathbf{L}_E^k(\boldsymbol{\mu}))_{nm} := \int_{\Omega} \varphi_n L_E^k(\boldsymbol{\mu})[\varphi_m], \quad (20)$$

$$(\mathbf{b}^k(\boldsymbol{\mu}))_n := \int_{\Omega} \varphi_n b^k(\boldsymbol{\mu}). \quad (21)$$

Then $u_N(\mathbf{x}, t; \boldsymbol{\mu})$ denotes a Reduced Basis (RB) approximation, which is defined to coincide with $u_N^k(\boldsymbol{\mu})$ in the time-slab $t \in [t_k, t_{k+1})$.

The well-definedness of the RB-approximation is valid under the assumptions on the operators of the evolution scheme.

Lemma 3.3 (Well-Definedness of RB-Approximation). *Given a parametrized evolution scheme and a RB-space \mathcal{W}_N , the RB-approximation u_N is unique. In particular, the explicit/implicit form in Def. 3.2 is a well-defined construction algorithm.*

Proof. The initial data coefficient vector \mathbf{a}^0 is obviously well defined by (18). By induction it is sufficient to argue, that the RB evolution equation (19) is well defined, i.e. a unique \mathbf{a}^{k+1} can be found for given \mathbf{a}^k . If $\mathbf{L}_I^k(\boldsymbol{\mu})$ was not regular, we could find linearly dependent rows, i.e. $\boldsymbol{\lambda} = (\lambda_n)_1^N \neq \mathbf{0}$ with $0 = \sum_n \lambda_n \langle \varphi_n, L_I^k[\varphi_m] \rangle$ for all m . Multiplying with λ_m and summation over m yields $0 = \langle \sum_n \lambda_n \varphi_n, L_I^k[\sum_m \lambda_m \varphi_m] \rangle$. The positive definiteness of $L_I^k(\boldsymbol{\mu})$ implies that $\sum_n \lambda_n \varphi_n = \mathbf{0}$, which contradicts to the linear independence of the basis-vectors in Φ_N . \square

4 A-Posteriori Error Estimates

The main goal of RB-methods in addition to the efficient approximation method itself is that they aim at tight a-posteriori error estimates, which enable to estimate the deviation $u_N - u_H$ from the unavailable high-resolution solution u_H during the online-phase. We can explicitly derive error bounds, which are based on the following representation of the residuals.

Lemma 4.1 (Residual Decomposition). *Let $u_N(\boldsymbol{\mu})$ denote the RB-approximation of a parametrized evolution scheme. For $k = 0, \dots, K - 1$ we introduce the residuals*

$$R^{k+1}(\boldsymbol{\mu}) := \frac{1}{\Delta t_k} \left(L_I^k(\boldsymbol{\mu})[u_N^{k+1}] - L_E^k(\boldsymbol{\mu})[u_N^k] - b^k(\boldsymbol{\mu}) \right). \quad (22)$$

Then the L^2 -norms of the residuals can be efficiently computed by

$$\|R^{k+1}(\boldsymbol{\mu})\|^2 = \frac{1}{(\Delta t_k)^2} \left((\mathbf{a}^{k+1})^T \mathbf{K}_{II}^k \mathbf{a}^{k+1} - 2(\mathbf{a}^{k+1})^T \mathbf{K}_{IE}^k \mathbf{a}^k \right. \quad (23)$$

$$\left. + (\mathbf{a}^k)^T \mathbf{K}_{EE}^k \mathbf{a}^k + m^k - 2(\mathbf{a}^{k+1})^T \mathbf{m}_I^k + 2(\mathbf{a}^k)^T \mathbf{m}_E^k \right) \quad (24)$$

with corresponding matrices, vectors and scalars

$$\left(\mathbf{K}_{II}^k(\boldsymbol{\mu}) \right)_{nm} := \int_{\Omega} L_I^k[\varphi_n] L_I^k[\varphi_m], \quad \left(\mathbf{K}_{IE}^k(\boldsymbol{\mu}) \right)_{nm} := \int_{\Omega} L_I^k[\varphi_n] L_E^k[\varphi_m], \quad (25)$$

$$\left(\mathbf{K}_{EE}^k(\boldsymbol{\mu}) \right)_{nm} := \int_{\Omega} L_E^k[\varphi_n] L_E^k[\varphi_m], \quad m^k(\boldsymbol{\mu}) := \int_{\Omega} b^k(\boldsymbol{\mu})^2, \quad (26)$$

$$\left(\mathbf{m}_I^k(\boldsymbol{\mu}) \right)_n := \int_{\Omega} L_I^k[\varphi_n] b^k(\boldsymbol{\mu}), \quad \left(\mathbf{m}_E^k(\boldsymbol{\mu}) \right)_n := \int_{\Omega} L_E^k[\varphi_n] b^k(\boldsymbol{\mu}). \quad (27)$$

Proof. The statement follows from squaring (22) and applying linearity of the detailed evolution-operators and the definitions from Def. 3.2. \square

Based on this residual definition we present two error estimates with different assumptions, error-norms and resulting accuracies.

First we obtain a $L^2(\Omega)$ -error estimate for all timesteps, which in principle gives an estimate in the $L^\infty([0, T_{\max}], L^2(\Omega))$ -norm. It is applicable both in the parabolic and hyperbolic case, it can be applied for full explicit, semi-implicit or full implicit discretizations.

Proposition 4.2 (A-Posteriori L^2 -Error Estimate). *Let u_H denote the detailed solution of a parametrized evolution scheme and u_N be the corresponding RB-approximation for a given RB-space \mathcal{W}_N . We assume that the RB-space contains the initial data, $u_H^0(\boldsymbol{\mu}) \in \mathcal{W}_N$. Then, the $L^2(\Omega)$ -approximation error is bounded for all time-steps k by*

$$\|u_N^k(\boldsymbol{\mu}) - u_H^k(\boldsymbol{\mu})\| \leq \Delta_N^k(\boldsymbol{\mu}) \quad \text{with} \quad (28)$$

$$\Delta_N^k(\boldsymbol{\mu}) := \sum_{n=0}^{k-1} \Delta t_n \|R^{n+1}\| (C_E)^{k-1-n}. \quad (29)$$

Proof. We want to bound the error $e^k := u_N^k - u_H^k$. For $k = 0$ the assumption $u_H^0 \in \mathcal{W}_N$ implies the claimed bound as $\|e^0\| = 0$. The definition of the residual (22) and the exact evolution scheme (11)-(12) allow rewriting

$$\begin{aligned} L_I^k[e^{k+1}] &= L_I^k[u_N^{k+1}] - L_I^k[u_H^{k+1}] \\ &= \Delta t_k R^{k+1} + L_E^k[u_N^k] + b^k - L_I^k[u_H^{k+1}] \\ &= \Delta t_k R^{k+1} + L_E^k[u_N^k] - L_E^k[u_H^k]. \end{aligned}$$

Thus, the error satisfies an evolution equation

$$L_I^k[e^{k+1}] = \Delta t_k R^{k+1} + L_E^k[e^k]. \quad (30)$$

The representation of $L_I^k = Id + \Delta t_k \bar{L}_I^k$ and the pd-ness of \bar{L}_I^k imply $\|(L_I^k)^{-1}\| \leq 1$. With continuity of the operators we obtain

$$\|e^{k+1}\| \leq \|L_I^k\|^{-1} \left(\Delta t_k \|R^{k+1}\| + \|L_E^k\| \|e^k\| \right) \leq \Delta t_k \|R^{k+1}\| + C_E \|e^k\|.$$

Resolving this recursion yields the proposed bound (28)-(29). \square

Estimates for the error measured in energy-norms can be provided if coercivity of \bar{L}_I^k is assumed. These bounds can be more accurate than the previous, but they are not applicable to the hyperbolic case (vanishing diffusivity) or in case of pure explicit discretizations. The following is a generalization of the estimate in [10], where we additionally treat the explicit operator contributions and emphasize the applicability to different norms by introducing a parameter γ . This parameter can be chosen in applications to optimize the effectivity of the bounds. In particular by $\gamma = 0$ we obtain an estimate for the $L^2(\Omega)$ -error, for $\gamma = 1$ we obtain the norm as used in the given reference.

Proposition 4.3 (A-Posteriori Weighted Energy-Error Estimate). *Let u_H denote the detailed solution of a parametrized evolution scheme and u_N be the corresponding RB-approximation for a given RB-space \mathcal{W}_N . We assume that the RB-space contains the initial data, $u_H^0(\boldsymbol{\mu}) \in \mathcal{W}_N$. If the explicit operator bound satisfies $C_E \leq 1$ and the implicit operator contribution \bar{L}_I^k is coercive with coercivity constant $\alpha > 0$, then we define an operator dependent norm for functions $v = \{v^k\}_{k=1}^K \subset (\mathcal{W}_H)^K$ by*

$$|v|_I := \left(\sum_{k=0}^{K-1} \Delta t_k \left\langle v^{k+1}, \bar{L}_I^k[v^{k+1}] \right\rangle \right)^{1/2}.$$

With the abbreviation $C := ((1 - C_E^2)^{1/2} + 1)/2$ we define for any weight $\gamma \in [0, C^{-1})$ a spatio-temporal energy-norm¹ by

$$\|v\|_\gamma := \left(\|v^K\|^2 + \gamma |v|_I^2 \right)^{1/2}.$$

Then, the error can be bounded by

$$\|u_N(\boldsymbol{\mu}) - u_H(\boldsymbol{\mu})\|_\gamma \leq \bar{\Delta}_{N,\gamma}^K(\boldsymbol{\mu}) \quad \text{with} \quad (31)$$

$$\bar{\Delta}_{N,\gamma}^K(\boldsymbol{\mu}) := \left(\frac{1}{4\alpha C(1 - \gamma C)} \sum_{k=0}^{K-1} \Delta t_k \|R^{k+1}\|^2 \right)^{1/2}. \quad (32)$$

¹Note, that for $\gamma = 0$ it actually only is a seminorm.

Proof. The proof is a modification of [10]. As before, we start with the evolution equation for the error (30) Testing with e^{k+1} , applying the Cauchy-Schwarz and operator estimates yields

$$\langle e^{k+1}, L_I^k[e^{k+1}] \rangle \leq C_E \|e^{k+1}\| \|e^k\| + \Delta t_k \|R^{k+1}\| \|e^{k+1}\|.$$

We insert the claimed decomposition $L_I^k = Id + \Delta t_k \bar{L}_I^k$ and apply Young's inequality $|c||d| \leq \frac{1}{2\rho^2}c^2 + \frac{1}{2}\rho^2d^2$ with $\rho^2 = C_E^{-1} + \sqrt{C_E^{-2} - 1}$, $c = \|e^{k+1}\|$ and $d = \|e^k\|$ on the right hand side. Rearranging the terms yields with the definition of C

$$C \|e^{k+1}\|^2 - C \|e^k\|^2 + \Delta t_k \langle e^{k+1}, \bar{L}_I^k[e^{k+1}] \rangle \leq \Delta t_k \|R^{k+1}\| \|e^{k+1}\|. \quad (33)$$

Second application of Young's inequality with $\rho^2 = 2\alpha(1 - \gamma C)$, $c = \|R^{k+1}\|$ and $d = \|e^{k+1}\|$ and coercivity of \bar{L}_I^k allows to estimate the right hand term

$$\Delta t_k \|R^{k+1}\| \|e^{k+1}\| \leq \frac{\Delta t_k}{4\alpha(1 - \gamma C)} \|R^{k+1}\|^2 + \Delta t_k(1 - \gamma C) \langle e^{k+1}, \bar{L}_I^k[e^{k+1}] \rangle.$$

Combining this with (33) and division by C yields the inequality

$$\|e^{k+1}\|^2 - \|e^k\|^2 + \gamma \Delta t_k \langle e^{k+1}, \bar{L}_I^k[e^{k+1}] \rangle \leq \frac{\Delta t_k}{4\alpha(1 - \gamma C)C} \|R^{k+1}\|^2.$$

Summation over $k = 0, \dots, K - 1$ results in

$$\|e^K\|^2 - \|e^0\|^2 + \gamma |e_I^2| \leq \frac{1}{4\alpha(1 - \gamma C)C} \sum_{k=0}^{K-1} \Delta t_k \|R^{k+1}\|^2.$$

As $e^0 \equiv 0$, this is the claimed bound (31)-(32). \square

Some remarks on these results are in order. First, both are a rigorous means for error control which are completely independent of the dimensionality H as long as the μ -dependent quantities (25)-(27), C_E and α are available. The error can be monitored during RB-simulation and used for predicting reliable accuracies or selecting basis-functions as applied in the next section.

Note, that the growing of the first bound Δ_N^k with k is not exponential, but at most linear if $C_E \leq 1$ and the residual norms remain bounded. The growing of the second bound $\bar{\Delta}_{N,\gamma}^k$ is of order $k^{1/2}$, so asymptotically, this bound is more tight. However, the estimates are useful in different parameter domains. The former is informative for non-asymptotic time-ranges and the hyperbolic or weakly parabolic case. As soon as the coercivity of the problem is large, the latter estimate is accurate. As both estimators are cheap to compute during simulation, the minimum of both can be taken as a combined L^2 -error estimator, i.e. $\tilde{\Delta}_N^k := \min(\Delta_N^k, \bar{\Delta}_{N,\gamma}^k)$ for $\gamma = 0$.

The last remark addresses the treatment of the initial data: The error estimates emphasize the need for good approximation of the initial data. If the initial data is not contained in \mathcal{W}_N , additional error terms are appearing in the error bounds, which makes them unnecessarily larger and dependent on H . This inhibits an offline/online decomposition as described in the next sections. Therefore, inclusion of the initial data $P[u_0(\mu)]$ in the RB-space is beneficial and still enables variation of the initial data if it is affinely parameter dependent as explained in the next section.

5 Offline/Online Decomposition

The main goal of RB-methods is to provide a fast simulation-phase, where multiple new, formerly unseen parameter choices μ are given, and fast simulation output including error bounds are required. Ideally, the online RB-simulation is completely independent of the dimensionality H of the detailed discretization space \mathcal{W}_H .

For time-independent scenarios (i.e. the operators in the evolution scheme are k -independent), the method up to now is already largely satisfactory, as only few operations depend on H : For a given reduced basis and a new parameter $\boldsymbol{\mu}$, the H -dependent but time-independent quantities (18), (20)-(21) and (25)-(27) are computed once before the RB-simulation and $\alpha(\boldsymbol{\mu})$, $C_E(\boldsymbol{\mu})$ are determined. The approximate time evolution itself can be performed H -independently, i.e. all quantities (23)-(24), (29) and (32) can be computed very efficiently.

The RB-methodology provides means for even resolving this marginal H -dependence and treating time-dependent problems. This is obtained by an offline/online decomposition, where the reduced basis and all H -dependent quantities are computed offline, while in the online phase, i.e. given a new $\boldsymbol{\mu}$, all computations are completely independent of H . This has been realized in RB-FEM-approaches as mentioned in the introduction, now an analogous decomposition can be performed in the case of parametrized evolution schemes.

For this we assume an affine parameter dependence of all data functions, as done for elliptic or parabolic problems, e.g. [23, 10].

Definition 5.1 (Affine Parameter Decomposition of Data). *The data functions of an advection-diffusion-problem allow an affine parameter decomposition, if they can be decomposed into finite sums of products of parameter-independent and space-independent functions:*

$$\mathbf{v}(\mathbf{x}, t; \boldsymbol{\mu}) = \sum_{q=1}^{Q_v} \mathbf{v}^q(\mathbf{x}, t) \sigma_v^q(\boldsymbol{\mu}), \quad d(\mathbf{x}, t; \boldsymbol{\mu}) = \sum_{q=1}^{Q_d} d^q(\mathbf{x}, t) \sigma_d^q(\boldsymbol{\mu}), \quad (34)$$

$$u_0(\mathbf{x}; \boldsymbol{\mu}) = \sum_{q=1}^{Q_{u_0}} u_0^q(\mathbf{x}) \sigma_{u_0}^q(\boldsymbol{\mu}), \quad b_{\text{dir}}(\mathbf{x}, t; \boldsymbol{\mu}) = \sum_{q=1}^{Q_{b_{\text{dir}}}} b_{\text{dir}}^q(\mathbf{x}, t) \sigma_{b_{\text{dir}}}^q(\boldsymbol{\mu}) \quad (35)$$

$$\text{and } b_{\text{neu},r}(\mathbf{x}, t; \boldsymbol{\mu}) = \sum_{q=1}^{Q_{b_{\text{neu},r}}} b_{\text{neu},r}^q(\mathbf{x}, t) \sigma_{b_{\text{neu},r}}^q(\boldsymbol{\mu}) \quad \text{for } r = 0, 1. \quad (36)$$

Note that this is not a severe restriction, as any smooth function can be approximated in this way, e.g. by Taylor expansion or more sophisticated *empirical interpolation* [3]. It is not crucial, that the coefficient functions $\sigma_{(\cdot)}^q$ are constant in time, this is only assumed for simplifying the presentation.

Similar to the decomposition of the data, we assume an affine decomposition for the evolution operators:

Definition 5.2 (Affine Parameter Decomposition of Parametrized Evolution Scheme). *The parametrized evolution scheme allows an affine parameter decomposition if it can be written as*

$$P[u_0(\boldsymbol{\mu})] = \sum_{q=1}^{Q_P} P[u_0]^q \sigma_P^q(\boldsymbol{\mu}), \quad L_I^k(\boldsymbol{\mu})[u] = \sum_{q=1}^{Q_{L_I}} L_I^{k,q}[u] \sigma_{L_I}^q(\boldsymbol{\mu}), \quad (37)$$

$$b^k(\mathbf{x}, \boldsymbol{\mu}) = \sum_{q=1}^{Q_b} b^{k,q}(\mathbf{x}) \sigma_b^q(\boldsymbol{\mu}), \quad L_E^k(\boldsymbol{\mu})[u] = \sum_{q=1}^{Q_{L_E}} L_E^{k,q}[u] \sigma_{L_E}^q(\boldsymbol{\mu}). \quad (38)$$

It is not difficult to satisfy these requirements, as the properties are already valid for the data functions. In particular the first requirement is satisfied with $P[u_0]^q = P[u_0^q]$ and $\sigma_P^q = \sigma_{u_0}^q$ due to the affine decomposition of the data functions. For the remaining terms, the numerical flux and the boundary discretizations must be chosen to similarly maintain these properties. For instance, the Lax-Friedrichs flux again satisfies these properties, whereas upwind fluxes in general do not and are subject of ongoing investigations for nonlinear schemes. It can easily be concluded from Prop. 2.4, that the FV-discretization of the advection-diffusion equation with fluxes (8)-(10) allows an affine decomposition in the sense of Def. 5.2.

The decomposition of the evolution scheme directly implies a corresponding decomposition for the RB-operators and auxiliary quantities for the error estimators. The following lemma is a direct consequence of the previous definitions.

Lemma 5.3 (Affine Decomposition of the RB-Scheme). *If the parametrized evolution scheme allows an affine parameter decomposition, the RB-scheme can similarly be decomposed as*

$$\mathbf{a}^0(\boldsymbol{\mu}) = \sum_{q=1}^{Q_{u_0}} \mathbf{a}^{0,q} \sigma_P^q(\boldsymbol{\mu}), \quad \mathbf{L}_I^k(\boldsymbol{\mu}) = \sum_{q=1}^{Q_{L_I}} \mathbf{L}_I^{k,q} \sigma_{L_I}^q(\boldsymbol{\mu}), \quad (39)$$

$$\mathbf{b}^k(\boldsymbol{\mu}) = \sum_{q=1}^{Q_b} \mathbf{b}^{k,q} \sigma_b^q(\boldsymbol{\mu}), \quad \mathbf{L}_E^k(\boldsymbol{\mu}) = \sum_{q=1}^{Q_{L_E}} \mathbf{L}_E^{k,q} \sigma_{L_E}^q(\boldsymbol{\mu}), \quad (40)$$

with definition of $\mathbf{a}^{0,q}$, $\mathbf{L}_I^{k,q}$, $\mathbf{L}_E^{k,q}$, $\mathbf{b}^{k,q}$ analogous to (18) and (20)-(21). Similarly, the quantities required for the error estimate can be decomposed as

$$\mathbf{K}_{II}^k(\boldsymbol{\mu}) = \sum_{q,q'=1}^{Q_{L_I}} \mathbf{K}_{II}^{k,q,q'} \sigma_{L_I}^q(\boldsymbol{\mu}) \sigma_{L_I}^{q'}(\boldsymbol{\mu}), \quad (41)$$

$$\mathbf{K}_{IE}^k(\boldsymbol{\mu}) = \sum_{q=1}^{Q_{L_I}} \sum_{q'=1}^{Q_{L_E}} \mathbf{K}_{IE}^{k,q,q'} \sigma_{L_I}^q(\boldsymbol{\mu}) \sigma_{L_E}^{q'}(\boldsymbol{\mu}), \quad (42)$$

$$\mathbf{K}_{EE}^k(\boldsymbol{\mu}) = \sum_{q,q'=1}^{Q_{L_E}} \mathbf{K}_{EE}^{k,q,q'} \sigma_{L_E}^q(\boldsymbol{\mu}) \sigma_{L_E}^{q'}(\boldsymbol{\mu}), \quad (43)$$

$$\mathbf{m}_I^k(\boldsymbol{\mu}) = \sum_{q=1}^{Q_{L_I}} \sum_{q'=1}^{Q_b} \mathbf{m}_I^{k,q,q'} \sigma_{L_I}^q(\boldsymbol{\mu}) \sigma_b^{q'}(\boldsymbol{\mu}), \quad (44)$$

$$\mathbf{m}_E^k(\boldsymbol{\mu}) = \sum_{q=1}^{Q_{L_E}} \sum_{q'=1}^{Q_b} \mathbf{m}_E^{k,q,q'} \sigma_{L_E}^q(\boldsymbol{\mu}) \sigma_b^{q'}(\boldsymbol{\mu}), \quad (45)$$

$$m^k(\boldsymbol{\mu}) = \sum_{q,q'=1}^{Q_b} m^{k,q,q'} \sigma_b^q(\boldsymbol{\mu}) \sigma_b^{q'}(\boldsymbol{\mu}), \quad (46)$$

with definitions of the $\mathbf{K}_{II}^{k,q,q'}$, $\mathbf{K}_{IE}^{k,q,q'}$, $\mathbf{K}_{EE}^{k,q,q'}$, $\mathbf{m}_I^{k,q,q'}$, $\mathbf{m}_E^{k,q,q'}$, $m^{k,q,q'}$ analogous to (25)-(27).

With these decompositions the offline/online procedure and herewith the goal of fast online phase for varying $\boldsymbol{\mu}$ can be realized. The price for this fast online stage is the offline-stage, in which auxiliary quantities are computed. This can be both time and memory intensive. We will quantify this in the experiments section.

5.1 Computational steps of the offline-phase

Given a reduced basis Φ_N , the offline computation consists of the following $\boldsymbol{\mu}$ -independent steps, which involve expensive operations of complexity H :

1. Data Components \implies Evolution Scheme Components:

Based on the affine decomposition components of the data $(\mathbf{v}^q, d^q, b_{\text{dir}}^q, u_0^q, b_{\text{neu},1}^q, b_{\text{neu},0}^q)$, the affine decomposition components of the evolution operators $(L_I^{k,q}, L_E^{k,q})$ and functions $b^{k,q}$, $P[u_0]^q$ are computed.

2. Evolution Scheme Components \implies RB-Scheme Components:

Based on the evolution components $(L_I^{k,q}, L_E^{k,q}, b^{k,q}, P[u_0]^q)$, the RB-scheme components $(\mathbf{L}_I^{k,q}, \mathbf{L}_E^{k,q}, \mathbf{b}^{k,q}, \mathbf{a}^{0,q})$ are computed.

3. Evolution Scheme Components \implies Error Estimator Components:

Based on the evolution components $(L_I^{k,q}, L_E^{k,q}, b^{k,q}, P[u_0]^q)$, the components for the error estimation $(\mathbf{K}_{II}^{k,q,q'}, \mathbf{K}_{IE}^{k,q,q'}, \mathbf{K}_{EE}^{k,q,q'}, \mathbf{m}_I^{k,q,q'}, \mathbf{m}_E^{k,q,q'}, m^{k,q,q'})$ are computed.

5.2 Computational steps of the online-phase

Given the precomputed quantities from the offline phase and a new parameter $\boldsymbol{\mu}$, the following steps are performed without any need of operations of complexity H :

1. Data Coefficients \implies Evolution Scheme Coefficients:

Based on the affine decomposition coefficients of the data $(\sigma_v^q(\boldsymbol{\mu}), \sigma_d^q(\boldsymbol{\mu}), \sigma_{b_{\text{dir}}}^q(\boldsymbol{\mu}), \sigma_{u_0}^q(\boldsymbol{\mu}), \sigma_{b_{\text{neu},1}}^q(\boldsymbol{\mu}), \sigma_{b_{\text{neu},0}}^q(\boldsymbol{\mu}))$, the affine decomposition coefficients of the evolution operators $(\sigma_{L_I}^q(\boldsymbol{\mu}), \sigma_{L_E}^q(\boldsymbol{\mu}))$ and functions $(\sigma_b^q(\boldsymbol{\mu}), \sigma_P^q(\boldsymbol{\mu}))$ are computed.

2. Evolution Scheme Coefficients \implies RB-Scheme Assembling:

Based on the evolution coefficients, the RB-operators and functions $(\mathbf{a}^0(\boldsymbol{\mu}), \mathbf{L}_I^k(\boldsymbol{\mu}), \mathbf{L}_E^k(\boldsymbol{\mu}), \mathbf{b}^k(\boldsymbol{\mu}))$ are assembled following (39)-(40).

3. Evolution Scheme Coefficients \implies Error Estimator Assembling:

Based on the evolution coefficients the auxiliary quantities for the error estimation $(\mathbf{K}_{II}^k(\boldsymbol{\mu}), \mathbf{K}_{IE}^k(\boldsymbol{\mu}), \mathbf{K}_{EE}^k(\boldsymbol{\mu}), \mathbf{m}_I^k(\boldsymbol{\mu}), \mathbf{m}_E^k(\boldsymbol{\mu}), m^k(\boldsymbol{\mu}))$ are assembled following (41)-(46).

Based on these components, the RB-online simulation (18)-(19), residual computation (23)-(24) and error estimation according to (29) and/or (32) can be performed.

6 Reduced Basis Construction

The prerequisite of the preceding sections was the availability of a reduced basis Φ_N . This choice determines the quality of the RB-simulation and error estimates, which means the algorithmic efficiency and numerical accuracy. Too few basis-vectors cannot guarantee a good solution and small predictive error, too many basis-vectors will deteriorate the computational complexity for the online computations.

We want to propose a method for incremental basis-construction (or incremental basis-refinement) by improving on an approach proposed in [10]. For simplification of presentation, we restrict ourselves to the L^2 -error and error estimator Δ_N^k . The method transfers identically to energy-error and estimators. The desirable goal is to construct for any given tolerance $\epsilon > 0$ a basis Φ_N with preferably small N , such that for all $\boldsymbol{\mu} \in \mathcal{P}$ and all $k = 0, \dots, K$ holds $\|u_H^k - u_N^k\| \leq \epsilon$. Due to the lack of such uniform $\boldsymbol{\mu}$ -independent a posteriori estimates one confines to a weaker statement of guaranteeing this estimate uniformly for a previously selected finite subset $\boldsymbol{\mu} \in M \subset \mathcal{P}$. The general procedure of incremental basis-construction in the offline-phase [9, 10] can be divided into the following steps

1. compute a small initial reduced basis Φ_{N_0} of N_0 basis-vectors
2. set $\Phi_N := \Phi_{N_0}$ and $N := N_0$
3. determine all offline-simulation quantities for the current RB-basis Φ_N
4. for all $\boldsymbol{\mu} \in M$ determine the posterior error estimates $\Delta_N^k(\boldsymbol{\mu}), 0 \leq k \leq K$
5. select a $\boldsymbol{\mu}^*, k^*$ with $\Delta_N^{k^*}(\boldsymbol{\mu}^*) = \max_{\boldsymbol{\mu} \in M, 0 \leq k \leq K} \Delta_N^k(\boldsymbol{\mu})$
6. perform the detailed simulation for parameter $\boldsymbol{\mu}^*$ resulting in $u_H(\boldsymbol{\mu}^*)$
7. determine a suitable new basis-function φ_{N+1} from this detailed simulation

8. generate the extended reduced basis $\Phi_{N+1} := \Phi_N \cup \{\varphi_{N+1}\}$
9. set $N := N + 1$ and repeat steps 3-9 until $\max_{\boldsymbol{\mu} \in M} \Delta_N^k(\boldsymbol{\mu}) \leq \epsilon$.

As the Δ_N^k are upper bounds for the true error, the resulting reduced basis in particular guarantees $\max_{\boldsymbol{\mu} \in M} \|u_H^k - u_N^k\|_{L^2(\Omega)} \leq \epsilon$. Note that in case of $C_E \geq 1$ the error estimates $\Delta_N^k(\boldsymbol{\mu})$ are monotonically increasing with k , and the maxima will be obtained at the final time-step K . This insight can be used for basis-construction as the maximum search can be simplified by choosing $k^* = K$.

Two components in this algorithm must be specified: Step 1, the initial choice of basis, and step 7, the rule for choice of the basis-extension vector.

Choice of Initial Basis

The insights on the a-posteriori error estimate already give a guideline for the choice of a good initial reduced basis: It should comprise all different possible initial data constellations $P[u_0(\boldsymbol{\mu})]$ for varying parameters in order to have the error estimate as small as possible. Due to the affine parameter dependence in (35), this is obtained by including the functions $P[u_0]^q$ for $q = 1, \dots, Q_{u_0}$ into the RB-space. Therefore, we choose Φ_{N_0} as an orthonormalization of these initial data components in step 1 of the basis-generation process.

Choice of Basis-Extension Method

For the extension of the reduced basis in step 7 of the basis-construction, two algorithms are used. The first one is chosen as a reference method, which is similar to the one applied in [10].

Basis-extension by maximum Δ -increase snapshot

Given that $\{\Delta_N^k(\boldsymbol{\mu}^*)\}_{k=0}^K$ is a monotonically increasing sequence, a large estimator increase from Δ_N^{k-1} to Δ_N^k is taken as an indicator, that the corresponding detailed solution u_H^k is not sufficiently accurate. Thus, the next vector φ_{N+1} is chosen by selecting the time index $k^* := \arg \max_k (\Delta_N^k(\boldsymbol{\mu}^*) - \Delta_N^{k-1}(\boldsymbol{\mu}^*))$ and orthonormalizing the snapshot $u_H^{k^*}(\boldsymbol{\mu}^*)$ with respect to the given basis Φ_N .

This algorithm has certain disadvantages as already partially mentioned in [9]: Only a single snapshot from the whole time-evolution is used in the new basis, which is a large loss of information and expensive computation time. Further, the assumption of estimator reduction by including the snapshot turns out to be wrong. It can happen, that the selected snapshot already is in the current RB-space, the estimator increase is simply due to that u_N^{k-1} is already an inaccurate approximation of u_H^{k-1} . To prevent an occasional infinite loop of repeatedly selecting the same snapshot for basis-extension an additional test and a *backup procedure* has to be included in the above extension method [9].

Therefore, we propose a new selection method by using the whole detailed time evolution $\{u_H(\boldsymbol{\mu}^*)\}_{k=0}^K$, instead of only selecting one snapshot. Further, we want to avoid getting stuck in an infinite loop of falsely chosen new basis-functions, which are already in the current RB-space. Lastly, we want to perform information compression in order to have an as compact basis as possible. These points can be realized by introducing procedures known from the model reduction technique of *proper orthogonal decomposition*, see for instance [16]. In particular we apply a version of the Karhunen-Loève transformation or Principal Component Analysis (PCA) for data compression, which respects initially defined fixspaces. Instead of solving the PCA via an expensive or infeasible $H \times H$ eigenvalue problem, it can be performed based on the $N \times N$ *empirical Gramian*. For computational details we refer to control theory literature [25], which denotes it the *method of snapshots* or to machine-learning literature, which denotes it as kernel-PCA, cf. [27].

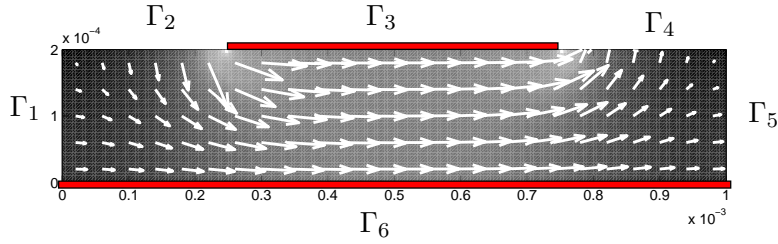


Figure 1: Illustration of the geometry and velocity field.

Basis-extension by PCA with fixspace

The trajectory $\{u_H^k\}_{k=0}^K$ is projected on \mathcal{W}_N^\perp and the projected sequence is subject to a PCA. The principal component, i.e. the vector of maximum variance is chosen as φ_{N+1} , which is readily orthonormal to Φ_N due to construction.

The latter choice of basis-extension vector ensures, that the maximum trajectory information is used, as φ_{N+1} represents the optimal new variance direction. Thus, a very compact basis is generated as demonstrated by the experiments

7 Experiments

The experiments have been realized in a MATLAB-environment ranging from detailed FV-simulations to full offline/online decomposition leading to an interactive online-demo, where parameter variation by the user is possible with instant RB-simulation result and error estimation.

Model Problem

The geometry of the model problem is illustrated in Fig. 1 and represents a simplified section of a gas-diffusion layer (gdl) of a fuel cell $\Omega = [0, 1 \cdot 10^{-3}] \times [0, 2 \cdot 10^{-4}]$ with impermeable boundaries Γ_3, Γ_6 at the middle of the top and the bottom, a gas inlet Γ_2 at the left top, a gas-outlet Γ_4 at the right top connecting the gdl to gas channels, and the left and right boundaries Γ_1 and Γ_5 leading to identical gdl-sections. The space discretization is a cartesian grid of 40×200 cells, the time range is $t \in [0, T_{\max} = 0.5]$ discretized with $K = 200$ equally sized time-intervals.

A time-independent velocity field is obtained from an elliptic finite element simulation for the pressure on an enlarged domain with a pressure decrease between the inlet and outlet. The negative of the resulting pressure gradient is taken as velocity field $\mathbf{v}(\mathbf{x})$ indicated with arrows in Fig. 1. The diffusivity is taken as $d(\mathbf{x}, t) \equiv k$ with parameter k . For realizing some variety in the resulting dynamics for different parameters, we consider initial data $u_0(\mathbf{x}) = \frac{1}{2}c_{\text{init}}(\sin(10000\pi x) + 1)$ with a parameter c_{init} interpolating between homogeneous zero initial data and the full sine-wave. The Dirichlet boundary values are set as $b_{\text{dir}}(\mathbf{x}, t) = \beta\chi_{\Gamma_2} + (1 - \beta)\chi_{\Gamma_4}$, where χ_{Γ_i} denote the indicator functions of the corresponding boundary segments. Thus, b_{dir} is parametrized by β , which models concentration differences between the inlet and outlet. The left boundary is an inflow boundary set to Dirichlet-value zero, the right boundary is an outflow boundary condition, the middle top and lower boundary are assigned noflow conditions, i.e. the Neuman boundary values are chosen as $b_{\text{neu}} = \chi_{\Gamma_5}(\mathbf{v}u) \cdot \mathbf{n}$.

By this we have specified our P²DE (1)-(4) with parameter vector $\boldsymbol{\mu} = (c_{\text{init}}, \beta, k)^T$ being variable in the range $\mathcal{P} := [0, 1] \times [0, 1] \times [0, 5 \cdot 10^{-8}]$. The solution variation with different parameter choices are visualized in Fig. 2. At initial time $t = 0$, the variation of the initial data is shown for $c_{\text{init}} = 0$ and $c_{\text{init}} = 1$ in the upper row. At some intermediate time $t = 0.1$, the different effects of diffusivity choice are visualized in the middle row for $k = 0$ and $k = 5 \cdot 10^{-8}$. The lower

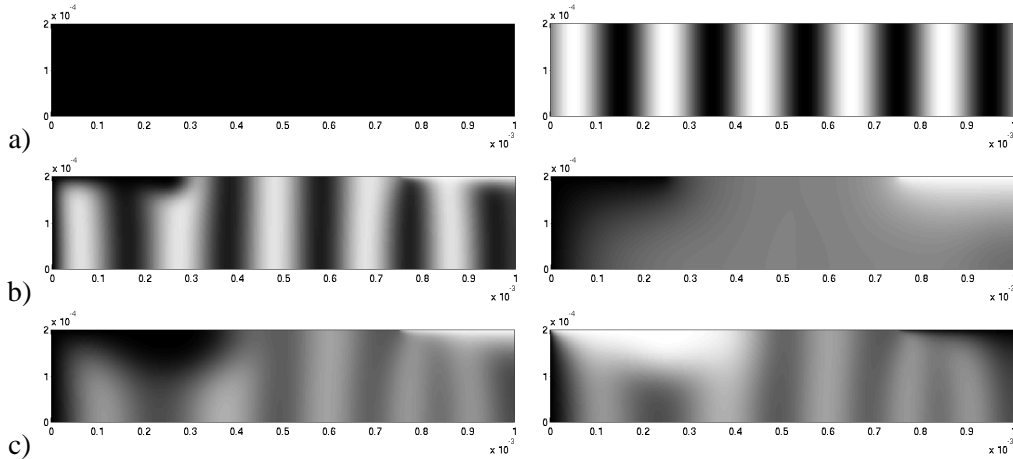


Figure 2: Illustration of solution variety by parameter variation. a) Initial data variation at start time, b) diffusion variation at intermediate time, c) inlet variation at end time.

row illustrates the effect of the different inlet/outlet concentrations by $\beta = 0$ and $\beta = 1$ at the end time $t = 0.5$.

Detailed versus Reduced Simulation

We perform a complexity comparison of the detailed FV-simulation versus the RB-simulation with 20 basis-functions, which is already a very good approximation if the basis-functions are chosen suitably, for instance as described in Sec. 6. We do not focus on accuracy of the RB-simulation in this section, the emphasis is put on memory and speed demands.

Concerning the runtime complexity, the main determining factors are the number of time-steps K , space-dimensionality H and number of reduced basis vectors N . Table 1 presents the results of the experimental assessments on a standard PC (Pentium 4, 2.8 GHz, 2 GB RAM) for two different space-time discretizations. The left part illustrates the general situation, where the data functions (velocities, diffusivities, boundary conditions) are time-dependent, i.e. the offline-data construction and online-assembly are performed for all time-steps. The right part demonstrates the case, where the data is constant in time, so the operators, offline-data and online-assembly only have to be performed once and can be reused for all time-steps. We report the simulation times for the detailed simulation, the reduced basis online and offline contributions. Here the offline-computations consist of preparing the matrices and auxiliary quantities. The reduced basis is assumed to be generated in advance. Results for the basis construction are given in a subsequent section. We consider the general case of advection-diffusion ($k \neq 0$), where the semi-implicit discretization requires the solution of a linear system in each time-step. In the second case of pure advection ($k = 0$), the implicit contributions vanish and the simulations simplify to matrix-vector multiplications. The time-step restrictions in both cases are given by the advection term which results in identical K in the advection-diffusion and pure advection case.

We first focus on the upper half of the table. In the setting of time-dependent data, the online RB-simulation is about 6-9 times faster than the detailed FV-simulation. About 1/3rd of the detailed computation time is spent in solving the linear system in the advection-diffusion case. This is not a relevant contribution in the RB case, as the matrices are smaller and the dominating time is required for assembling of the online auxiliary quantities, which has almost identical complexity in the explicit or the semi-implicit case.

In the right part of the upper table, the case of data functions which are constant in time, all times are largely reduced. The RB-offline-phase is almost ideally sped up by factor $K = 200$. The complexity difference between the detailed and the RB-online-phase is more expressed here than in

Table 1: Time complexity comparisons between detailed and RB-simulation in the cases of advection-diffusion and pure advection and the data functions being constant in time or not.

Discretization: 40×200 cells, $K = 200$ timesteps						
	Data non const in time			Data const in time		
	Detailed	RB online	RB offline	Detailed	RB online	RB offline
advection-diffusion	155.94s	16.67s	447.16s	45.67s	1.02s	2.41s
advection	105.97s	16.53s	437.20s	1.51s	0.79s	2.31s

Discretization: 80×400 cells, $K = 1000$ timesteps						
	Data non const in time			Data const in time		
	Detailed	RB online	RB offline	Detailed	RB online	RB offline
advection-diffusion	4043.18s	143.57s	8693.90s	924.91s	6.18s	9.22s
advection	2758.20s	134.00s	8506.60s	17.41s	3.64s	8.83s

Table 2: Memory requirement [KB] with increasing number of basis-functions and time-dependent or time-independent data.

Discretization: 40×200 cells, $K = 200$ timesteps						
	Number of basis-functions					
	20	40	60	80	100	120
Reduced Basis	1250	2500	3750	5000	6250	7500
Offline data const in time	230	898	2002	3545	5525	7942
Offline data non const in time	46014	179452	400390	708827	1104765	1588203

the time-dependent case, the speedup-factors range from 2-45.

In the lower part of the table we repeat the experiment with a refined discretization in space (factor 2 per dimension) and time (factor 5) in order to give an impression of the scaling behaviour. The high number of timesteps is required due to higher velocities near the better resolved singularities at the interface of boundary segments Γ_2/Γ_3 and Γ_3/Γ_4 . The main qualitative observations are valid as before. However, the relative complexity gains are much better expressed, as the detailed simulations largely depend on the space resolution in contrast to the reduced online simulations. The speedup factors in the non-constant data setting range from 21-28 and in the constant data case from 5-150.

Overall, high speedup can be obtained, e.g. in case of implicit contributions, where the dominating time of the detailed simulation is the solution of the large linear systems. On the other hand, the RB-offline phase is very demanding, in particular scaling linearly both with H and K , which is clearly reflected in the factor 20 in the left part of the table. Remind, that this is no problem for the scenario of repeated RB-simulations with varying parameters, as only the online phase is relevant here.

Concerning memory requirements, the main factor again is the number of time-steps and the number of basis-functions as the required matrices scale quadratically with N and the vectors and reduced basis size grow linearly. For the implementation of the model problem with FV-discretization as given in the previous sections, the affine decompositions of the detailed operators are $Q_P = 1, Q_{L_I} = 4, Q_{L_E} = 5, Q_b = 7$. Correspondingly, the number of components of the RB-operators range from 4 to 7, the number of components for the error estimator range from 16 to 49. As the matrices in the offline and online-data are non-sparse in general, we obtain the memory requirements for the offline data for growing basis size as given in Tab. 2. The linear scaling in memory storage for the reduced basis and the quadratic scaling in the offline data is clearly visible.

In general, for low number of basis-functions, the memory requirements may be dominated by the detailed reduced basis vectors, but with increasing number of basis-functions, at some point

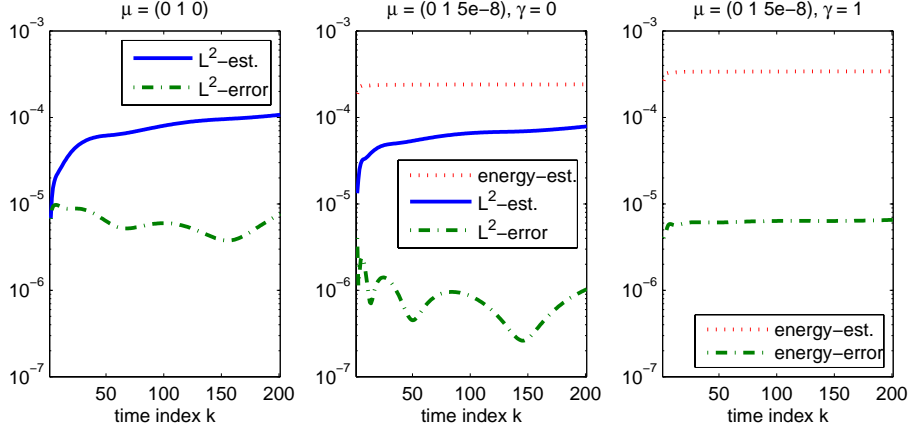


Figure 3: Time-behaviour of error estimators Δ_N^k and $\bar{\Delta}_{N,\gamma}^k$ versus true error $\|e^k\|$, $\| \|e^k\| \|_\gamma$. a) L^2 -error and estimator in case of no diffusivity, b) energy- and L^2 -estimator and L^2 -error for large diffusivity, c) energy-estimator and -error for large diffusivity.

the offline data requirements are the demanding and limiting factor. In the case of time-dependent data functions, this is even more expressed. The requirements mainly scale with the number of time-steps. With 200 time-steps in our case, 120 reduced basis vectors already imply 1.6 GB of offline data. This is not only a capacity problem, but also limiting the reduced basis simulation. This offline data must be accessed in total in each online simulation run during operator assembly. A solution for this problem can be based on the idea of keeping the reduced basis small [9]. As a small basis can only capture limited variability in time, only a limited time-interval can be modelled well with a small basis. Therefore, multiple reduced bases can be constructed, each suited to a certain time-interval.

Quality of Error Estimators

Reusing the RB-space spanned by the 20 basis-vectors as in the preceding section, we investigate the predictivity of the error estimators in Figs. 3 and 4. (L^2 -estimator Δ_N^k : blue, solid; energy-estimator $\bar{\Delta}_{N,\gamma}^k$: red, dotted) in comparison with the true error ($\|e^k\|$, $\| \|e^k\| \|_\gamma$: green, dashed-dotted). We choose a corner of the domain \mathcal{P} with diffusivity $k = 0$ and logarithmically plot the true L^2 -error $\|e^k(\boldsymbol{\mu})\| = \|u_H^k(\boldsymbol{\mu}) - u_N^k(\boldsymbol{\mu})\|_{L^2(\Omega)}$ versus the error estimator $\Delta_N^k(\boldsymbol{\mu})$ over time indexed by k in the first plot of Fig. 3. For diffusivity $k > 0$ the energy-estimators are defined, so we additionally plot this quantity for $\gamma = 0$ in the middle plot. The right plot illustrates the $\gamma = 1$ energy-error estimator and the corresponding energy-error.

The results uniformly reflect the conservativity of the error estimators by $\Delta_N^k(\boldsymbol{\mu}) \geq \|e^k(\boldsymbol{\mu})\|$ and $\bar{\Delta}_{N,\gamma}^k \geq \| \|e^k(\boldsymbol{\mu})\| \|_\gamma$. Due to our operator-bound $C_E = 1$, the estimators are monotonically increasing, whereas the true L^2 -error can be non-monotonic, depending, on whether the current state is captured well or inaccurate in the RB-space. In the present range of diffusivity and time index, the L^2 -estimators are preferable to the energy-estimates, as they are more tight. However, the slower asymptotic growing rate of the energy-estimators in time is also visible.

We want to illustrate the predictivity of the error estimators under varying parameters. Figure 4 presents some results of the quantities evaluated at the end time $t = T_{\max}$. The first plot demonstrates the L^2 -error estimator Δ_N^K and the true error at $\boldsymbol{\mu} = (c_{\text{init}} = 1, \beta, k = 0)$ under variation of β , the middle plot for $\boldsymbol{\mu} = (1, 0, k)$, additionally contains the energy-estimator $\bar{\Delta}_{N,\gamma}^k$ for $\gamma = 0$. The last plot illustrates the $\gamma = 1$ error estimator $\Delta_{N,\gamma}^K$ for $\boldsymbol{\mu} = (c_{\text{init}}, 1, 5 \cdot 10^{-8})$ and corresponding energy-error.

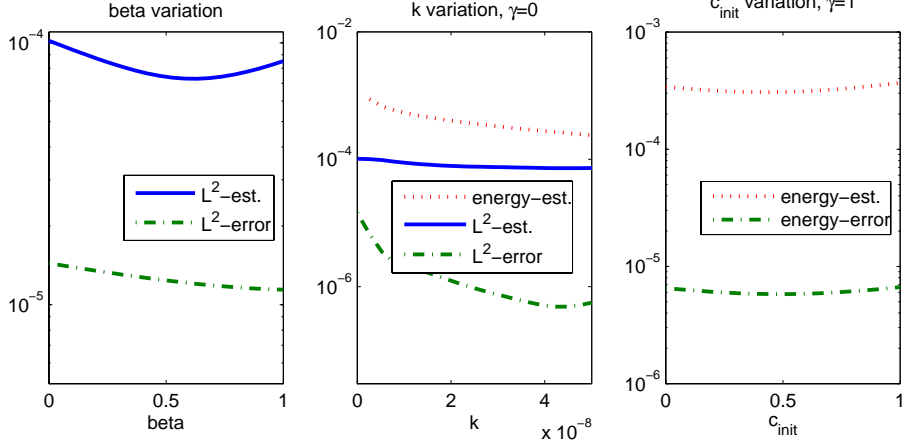


Figure 4: Predictivities of error estimators at $t = T_{\max}$ for parameter variations. a) Δ_N^k and $\|e^k\|$ for $c_{\text{init}} = 1$, $k = 0$ and variation of β , b) $\bar{\Delta}_{N,\gamma}^k$, Δ_N^k and $\|e^k\|_\gamma$ for $\gamma = 0$, $c_{\text{init}} = 1$, $\beta = 0$ and variation of k , c) $\bar{\Delta}_{N,\gamma}^k$ and $\|e^k\|_\gamma$ for $\gamma = 1$, $\beta = 1$, $k = 5 \cdot 10^{-8}$ and variation of c_{init} .

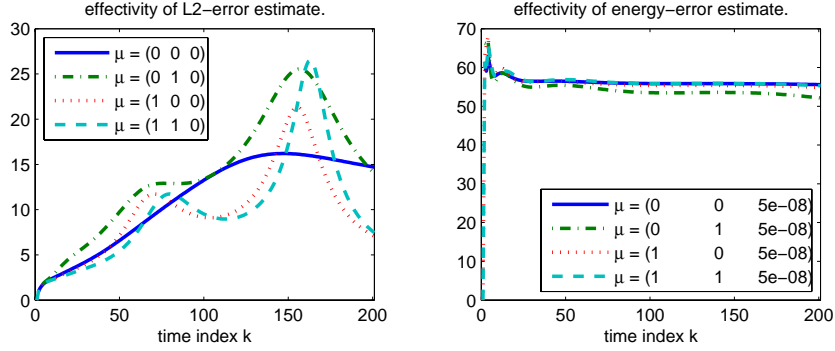


Figure 5: Effectivities of error estimators over time for different μ . a) Effectivity $\eta_N^k(\mu)$ of the L^2 -estimator $\Delta_N^k(\mu)$ for $k = 0$, b) Effectivity $\bar{\eta}_{N,\gamma}^k(\mu)$ of the energy-estimator $\bar{\Delta}_{N,\gamma}^k(\mu)$ with $\gamma = 1$ and $k = 5 \cdot 10^{-8}$.

In all plots the parameters μ with maximum error are correctly predicted by the estimators. This is an underpinning of incremental basis-construction algorithms, as indeed the parameters that are identified online are likely to be parameters with maximum true error. The middle plot once more clearly emphasizes the difference and applicability of the L^2 - and energy-error estimators: The former is useful in the whole parameter range covering hyperbolic and parabolic problems, the latter is only applicable in the parabolic case preferable for large diffusivities and times. As soon as the diffusivity goes to zero, the energy-estimators diverge and are not informative.

In addition to the conservativity and predictivity of the error estimators, we want to assess the effectivities $\eta_N^k(\mu) := \Delta_N^k(\mu) / \|e^k(\mu)\|$ and $\bar{\eta}_{N,\gamma}^k(\mu) := \bar{\Delta}_{N,\gamma}^k(\mu) / \|e^k(\mu)\|_\gamma$. Figure 5 gives the results of these quantities over time. Plot a) illustrates the effectivities for the L^2 -estimator in the hyperbolic case, i.e. $k = 0$, by plotting one curve for each of the four corresponding corners μ of the parameter cube \mathcal{P} . Plot b) depicts the effectivities of the energy-error estimators in the parabolic case $k = 5 \cdot 10^{-8}$ by similarly plotting one curve for each of the corresponding corners μ .

It is obvious, that the conservativity of the error estimator guarantees $\eta_N^k(\mu), \bar{\eta}_{N,\gamma}^k(\mu) \geq 1$. The factor of overestimation of the error is ranging from one to two orders of magnitude, which is a quite accurate prediction in addition to the large computational gain.

Reduced Basis Construction

In this last experimental section we report on results concerning the reduced basis construction with the algorithms given in Sec. 6. The estimator-threshold was set to $\epsilon = 1 \cdot 10^{-7}$ and a finite subset $M \subset \mathcal{P}$ consisting of an equidistant cartesian grid was chosen. Figure 6 illustrates the main findings. In plot a) the decrease of the maximum error estimator $\max_{\mu \in M} \Delta_N^K(\mu)$ is plotted with increasing basis size N . The set M consists of 5^3 parameters. It can be seen in the logarithmic scale for the error estimator, that both the PCA-fixspace and the max-error-snapshot extension methods are able to decrease the maximum error estimator exponentially in N .

Obviously, the PCA-based algorithm indeed is more efficient than the snapshot selection both in terms of final basis size and the computation time for assembling the basis: The PCA-fixspace algorithm obtains the desired threshold with 123 basis-functions in 6.1 hours, the snapshot-selection algorithm ends with 136 basis-functions in 7.4 hours. Concerning the offline-computation time, this is a 17.4% improvement in computation time and a 9.6% saving in basis size for given accuracy. Due to the squared memory requirements in the online-phase, this factor results in a 18.2% memory saving for the auxiliary matrices.

The set M can be understood as the training-set for the basis-generation. So more interesting than the error on the training-set, is a test-set accuracy by choosing a random set of new parameters M' with $M \cap M' = \emptyset$ and similarly determining $\max_{\mu \in M'} \Delta_N^K(\mu)$. This quantity for the PCA-fixspace algorithms is plotted in b) for a uniformly distributed random test-set M' of size 100 for different sets M used in basis-construction. Several curves are plotted corresponding to subsequent globally refined training-parameter sets M of size 3^3 , 5^3 and 9^3 . It is obvious in the $|M| = 5^3$ curve that the error estimates for test-parameters are up to two orders of magnitude larger than for training-parameters. The convergence in the test-error is not exponential as in the first plot, which indicates that the set M is chosen suboptimally. By refinement of the training-set M from size 3^3 over 5^3 to 9^3 , the test-error can be lowered by one order of magnitude, but still being far from the exponential decay of the training-error. So we conclude that methods of suitably and adaptively choosing the set M must be developed instead of choosing M a priori fixed.

The incremental basis-construction with both extension methods can result in one μ to be chosen as basis-extension parameter μ^* multiple times. For preventing subsequent multiple identical detailed simulation runs, these trajectories $\{u_H^k(\mu^*)\}_{k=0}^K$ could be cached temporarily until the basis-construction is finished. For our current implementation, however, the runtimes are acceptable without this caching strategy. The frequency of selecting any $\mu \in M$ as μ^* in the given experiment for the $|M| = 5^3$ cartesian grid is plotted in Fig. 7. It can be seen, that the automatic parameter selection is much more efficient than for instance selecting one basis-function for all parameters in M : Only a small subset of the 125 parameter vectors in M is selected for basis-construction. Most of the vectors are required for small diffusion coefficients k , whereas for large diffusivities, only few basis-vectors are required. This is in accordance with the intuition: For large diffusivities k , the end-time solutions of various choices of c_{init}, β are similar as detailed effects are smeared out. For small diffusivities k , the end-state $u_H^K(\mu)$ still has a high spacial variability with respect to the remaining parameters c_{init}, β , so more basis-vectors are required here.

For parameters, which effect in a linear superposition of data, e.g. c_{init} for the initial data or β for the inlet-parameter, the selected parameters μ^* tend to be at the borders of the intervals. For other parameters, such as the diffusivity k , the selected values can range over the whole allowed interval. This different quality of parameter dependence is reflected in the plot.

8 Discussion and Conclusion

We presented a formulation of the RB-methodology for arbitrary parametrized evolution schemes, which allow an explicit/implicit operator decomposition. We have demonstrated that our approach

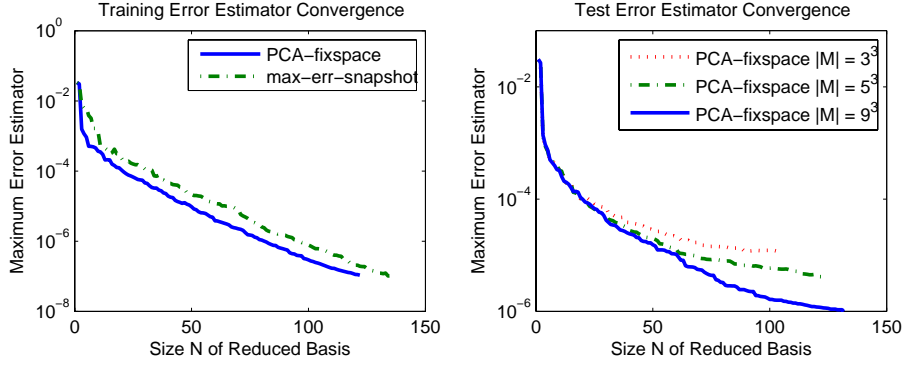


Figure 6: Illustration of basis-generation. a) Comparison of different algorithms by maximum training error estimator for μ chosen from the set M , b) maximum test-error estimator on a random test-set M' for different training-set sizes.

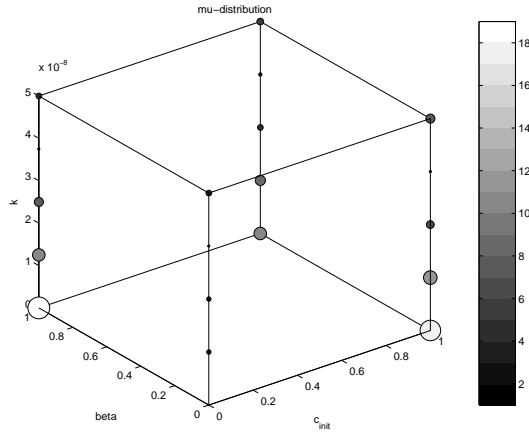


Figure 7: Frequency of μ^* -selection during reduced basis generation on a training-set $M \subset \mathcal{P}$ consisting of an equidistant cartesian grid of 5^3 points μ .

in particular covers FV-methods for parabolic and hyperbolic equations. Our operator formulation also covers finite element formulations. Thus, the RB-scheme formulation and error estimates can be seen an extension or alternative to existing approaches. Our operator formulation also covers more general schemes such as higher order schemes which include Runge-Kutta time discretizations, or local discontinuous Galerkin discretizations in space [5]. As long as the problem and the fluxes are linear and no limiters are involved, the methodology is directly transferable.

As an extension, estimates for output functionals can be derived instead of only considering the field variable. If, for instance, a linear functional output $s(u_H^k)$ is desired, this can be approximated by $s(u_N^k)$. The given L^2 -error-bound Δ_N^k directly implies a bound on the output-error by $|s(u_N^k) - s(u_H^k)| = |s(e^k)| \leq \|s\| \Delta_N^k$. More sophisticated error bounds might be derived with primal-dual techniques as they exist for the elliptic or parabolic case [23, 10].

As a second attractive perspective, the extension of the method to nonlinear FV- or local discontinuous Galerkin (LDG) schemes seems possible by affine approximations and collateral reduced basis spaces for the nonlinearity. This is currently under detailed investigation.

A further promising direction for future research is seen in the offline automatic basis-construction. We have illustrated that automatic basis-construction yields information about variability of the underlying physics: Parameter distributions during basis-generation indicate parameter ranges with high dynamics in time or with solution structure being distinct from other parameter ranges. E.g. large diffusivity coefficients produce similar solutions for wide ranges of the other parameters,

in contrast to small diffusivities. The empirical convergence analysis demonstrated that there is a big gap between the optimal exponential error decrease on the training-set of parameter values M and the error decrease on arbitrary test-parameters. If uniform convergence statements cannot be guaranteed, at least algorithms should be found, which adaptively construct reduced bases with empirical uniform exponential convergence.

9 Acknowledgement

The first author was funded by the German Federal Ministry of Education and Research under grant number 03SF031A. We thank Gianluigi Rozza for fruitful discussions on the subject.

References

- [1] B.O. Almroth, P. Stern, and F.A. Brogan. Automatic choice of global shape functions in structural analysis. *AIAA J.*, 16:525–528, 1978.
- [2] D.N. Arnold, F. Brezzi, B. Cockburn, and L.D. Marini. Unified analysis of discontinuous galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779, 2002.
- [3] M. Barrault, Y. Maday, N.C. Nguyen, and A.T. Patera. An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. *C. R. Acad. Sci. Paris Series I*, 339:667–672, 2004.
- [4] T. Barth and M. Ohlberger. Finite volume methods: Foundation and analysis. In E. Stein, R. de Borst, and T.J.R. Hughes, editors, *Encyclopedia of Computational Mechanics*. John Wiley & Sons, 2004.
- [5] B. Cockburn. Discontinuous Galerkin methods for computational fluid dynamics. In E. Stein, R. de Borst, and T.J.R. Hughes, editors, *Encyclopedia of Computational Mechanics*. John Wiley & Sons, 2004.
- [6] B. Cockburn and C.-W. Shu. Runge–Kutta discontinuous Galerkin methods for convection-dominated problems. *J. Sci. Comput.*, 16(3):173–261, 2001.
- [7] R. Eymard, T. Gallouët, and R. Herbin. Finite volume methods. In *Handbook of numerical analysis*, volume VII, pages 713–1020. North-Holland, Amsterdam, 2000.
- [8] E. Godlewski and P.-A. Raviart. *Numerical Approximation of Hyperbolic Systems of Conservation Laws*. Springer, 1996.
- [9] M.A. Grepl. *Reduced-basis Approximations and a Posteriori Error Estimation for Parabolic Partial Differential Equations*. PhD thesis, Massachusetts Institute of Technology, May 2005.
- [10] M.A. Grepl and A.T. Patera. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM-Math. Model. Num.*, 39(1):157–181, 2005.
- [11] R. Herbin and M. Ohlberger. A posteriori error estimate for finite volume approximations of convection diffusion problems. In *Proc. 3rd Int. Symp. on Finite Volumes for Complex Applications - Problems and Perspectives*, pages 753–760, 2002.
- [12] D. Kröner. *Numerical Schemes for Conservation Laws*. John Wiley & Sons and Teubner, 1997.
- [13] R.J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.
- [14] L. Machiels, Y. Maday, I.B. Oliveira, A. Patera, and D.V. Rovas. Output bounds for reduced-basis approximations of symmetric positive definite eigenvalue problems. *C. R. Acad. Sci. Paris Series I*, 331:153–158, 2000.

- [15] M. Mangold and M. Sheng. Nonlinear model reduction of a 2D MCFC model with internal reforming. *Fuel Cells*, 4(1–2):68–77, 2004.
- [16] B.C. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Trans. Automat. Control*, AC-26(1):17–32, 1981.
- [17] N.C. Nguyen, K. Veroy, and A.T. Patera. Certified real-time solution of parametrized partial differential equations. In S. Yip, editor, *Handbook of Materials Modeling*, pages 1523–1558. Springer, 2005.
- [18] A.K. Noor and J.M. Peters. Reduced basis technique for nonlinear analysis of structures. *AIAA J.*, 18(4):455–462, 1980.
- [19] M. Ohlberger. A posteriori error estimates for vertex centered finite volume approximations of convection-diffusion-reaction equations. *M2AN Math. Model. Numer. Anal.*, 35(2):355–387, 2001.
- [20] M. Ohlberger and J. Vovelle. Error estimate for the approximation of non-linear conservation laws on bounded domains by the finite volume method. *Math. Comp.*, 75:113–150, 2006.
- [21] T.A. Porsching and M.L. Lee. The reduced basis method for initial value problems. *SIAM J. Numer. Anal.*, 24(6):1277–1287, 1987.
- [22] C. Prud’homme, D. Rovas, K. Veroy, and A.T. Patera. A mathematical and computational framework for reliable real-time solution of parametrized partial differential equations. *ESAIM-Math. Model. Num.*, 36(5):747–771, 2002.
- [23] C. Prud’homme, D.V. Rovas, K. Veroy, L. Machiels, Y. Maday, A.T. Patera, and G. Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods. *J. Fluids Engineering*, 124:70–80, 2002.
- [24] A. Quarteroni, G. Rozza, L. Dede, and A. Quaini. Numerical approximation of a control problem for advection-diffusion processes. In *In System Modeling and Optimization, Proceedings of 22nd IFIP TC7 Conference*, 2006.
- [25] C.W. Rowley. Model reduction for fluids, using balanced proper orthogonal decomposition. *Int. J. Bifurcat. Chaos*, 15(3):997–1013, 2005.
- [26] G. Rozza. *Shape design by optimal flow control and reduced basis techniques: Applications to bypass configurations in haemodynamics*. PhD thesis, École Polytechnique Fédérale de Lausanne, November 2005.
- [27] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, 2002.
- [28] K. Veroy and A.T. Patera. Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: Rigorous reduced-basis a posteriori error bounds. *Int. J. Numer. Meth. Fluids*, 47:773–788, 2005.
- [29] K. Veroy, C. Prud’homme, and A.T. Patera. Reduced-basis approximation of the viscous Burgers equation: rigorous a posteriori error bounds. *C.R. Acad. Sci. Paris, Ser. I*, 337:619–624, 2003.